

## Proyecto Final: Ingeniería de Datos

### Integrantes:

- Juan Fernando Morales
- Sofia Marin Guacaneme

### Índice:

#### 1. Análisis y presentación de la problemática:

¿Qué queremos solucionar?

#### 2. Modelo Empleado

¿Cómo intentamos darle solución al proyecto?

#### 3. Resultados

¿Cómo solucionamos los principales tropiezos del cliente?

#### 4. Conclusiones

### Análisis:

- Problemática:

Hace 2 años entramos en una situación que llegó a cambiar la vida de todos en el mundo, la pandemia del covid 19 obligó a transformar los tradicionales de negocios (restaurantes, cines, comercios, gimnasios, etc) en modelos prácticos y adaptados a la distancia. Las ventas en línea fueron la salvación de la economía y bastantes personas tomaron la decisión de emprender gracias a las herramientas que 'facilitaban' el alcance de sus productos en línea, unos por necesidad u otros viendo la oportunidad de lograr ingresos de más desde la comodidad de la casa.

Unos de los grandes retos de los emprendedores que nacieron gracias al Covid fueron aprender a llevar un control de los ingresos y egresos, llevar un inventario exacto de todos sus productos y facilitar un catálogo práctico para sus clientes.
- Motivación:

Rose and Brave (R&B) es un emprendimiento de Bogotá que decide empezar durante la pandemia, R&B vende accesorios de excelente calidad y a muy buen precio, fue creado por una joven de 16 años que sin tener mayor idea de un control de negocios fue aprendiendo para sacarlo adelante.
- Cosas para tener en cuenta:
  - Entender las necesidades de un negocio tales como un registro de compradores, un control de mercancía, vigilancia de ingresos y egresos, repertorio de proveedores, entre otros.
- Propuesta Innovativa:

Un sistema que almacene información de tal forma que ayude al comerciante a llevar control de los artículos vendidos y por vender (disponibilidad), almacenar la información de cada cliente, saber las ganancias de cada pedido, conocer la cantidad de pedidos despachados, valor del envío dependiendo de el lugar donde se debe entregar, los clientes frecuentes, los datos de los proveedores y así mismo facilitar al cliente su experiencia de

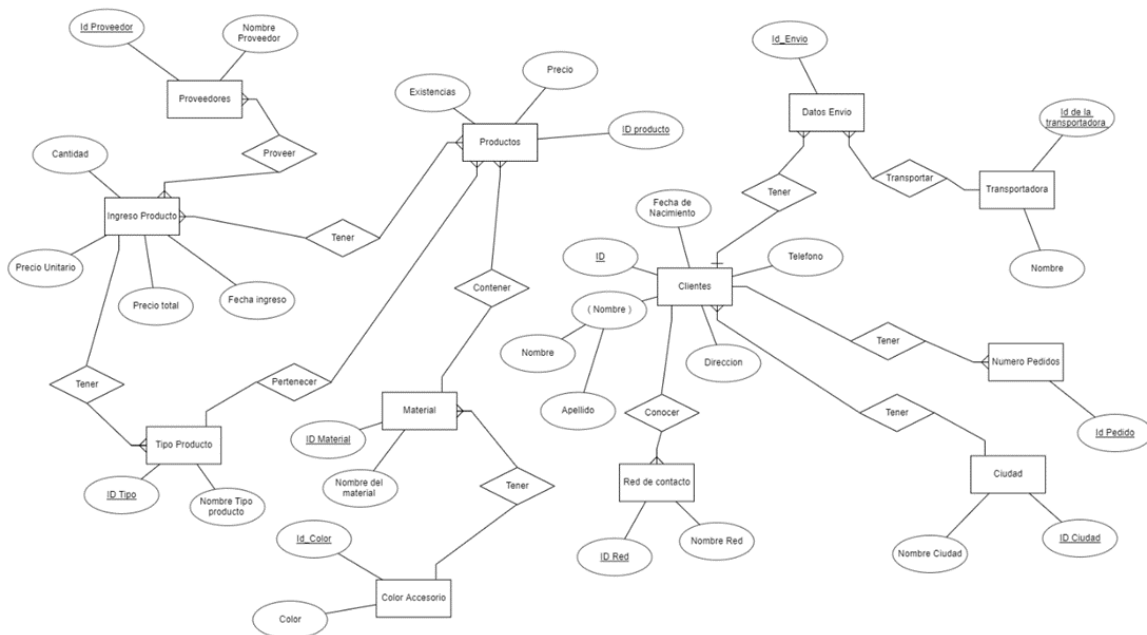
compra, como por ejemplo una búsqueda específica de lo que necesita, un control del envío de su pedido, entre otras.

- **Requerimientos de información:**

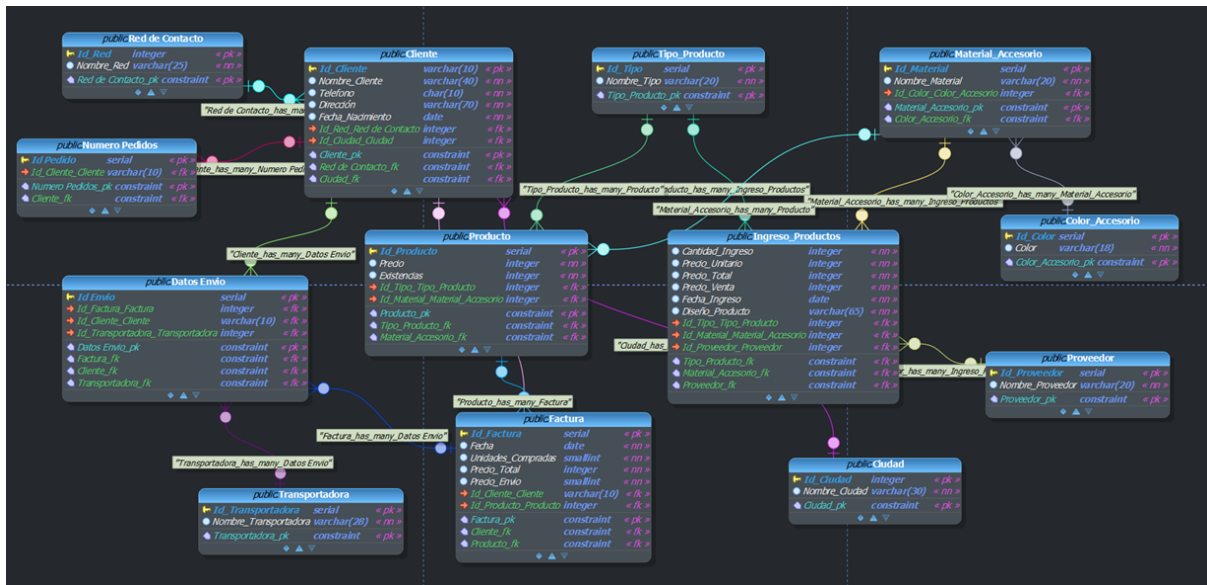
1. Conocer datos básicos de cada cliente (nombre, cédula, celular, dirección).
2. Tener en cuenta los tipos de accesorios que maneja la tienda y sus especificaciones (material, diseño, tamaño, etc).
3. Información sobre los proveedores (nombre, ciudad) y que suministra a la tienda cada uno.
4. Saber las transportadoras que maneja la tienda y valores dependiendo del lugar de entrega.

### Modelo Empleado:

Con ayuda del SQL decidimos implementar un sistema que recogiera de forma fácil toda la información que la dueña de este emprendimiento requiere para llevar de la mejor manera el servicio y orden de su local, para empezar hay que generar un diseño fácil de entender para que con ayuda del cliente, se nos permita reconocer todo lo que el modelo requiere solventar



Después de esto tenemos que identificar el método más óptimo para realizar el modelo, implementando recursos importantes del SQL: La normalización



Viendo lo anterior podemos ver el cambio que obtendrá la dueña de este negocio al final del proyecto, un sistema de facturas que en definitiva erradica todos los tropiezos que presentó sin la ayuda de este modelo. La factura le permite llevar el control de los clientes, los datos fundamentales de los clientes (teléfono, dirección, ciudad), conocer los productos que salen del stock de la tienda y así tener un catálogo al día, saber sus ingresos y egresos, sus ganancias también y hasta la visión de cómo se mueve su negocio

## Resultados:

Después de toda la creación podemos ver una toma de datos importante:

- Logramos construir consultas que nos facilitaran filtrar clientes o productos:

```
--Consultas

-- Productos con sus ventas

select f.id_producto, p.diseño_producto, count(f.id_producto)
from factura f, producto p
where f.id_producto = p.id_producto
group by f.id_producto, p.diseño_producto;

--Productos con menos unidades

select*
from producto p
where p.existencias <= 2;
```

- Funciones que nos ayudan por ejemplo a conocer los clientes más frecuentes en la tienda:

```
CREATE OR REPLACE FUNCTION mayor_comprador()
    RETURNS TABLE(
        nombre varchar,
        numero_ped integer,
        ranking bigint)
AS
$$
BEGIN
    RETURN QUERY
        (select cliente.nombre ,numero_pedidos,rank() over(ORDER BY numero_pedidos DESC)
         from numero_pedidos, cliente
         where numero_pedidos.id_cliente = cliente.id_cliente
         limit 1);
END;
$$
LANGUAGE plpgsql;
```

- Vistas que le permita al cliente buscar productos a su gusto, por ejemplo en un rango de precios:

```
-----
--Vistas
-----

CREATE VIEW producto_rangoprec(id_producto,precio)
AS
SELECT id_producto,precio
from producto
where 20000>precio and precio<10000;
```

- Creación de Roles para darle prioridad a la dueña del emprendimiento:

```
-----
--Roles
-----

CREATE ROLE Owner WITH LOGIN PASSWORD 'roseandbrave123'; --crear rol de dueño
REVOKE CREATE ON SCHEMA public FROM PUBLIC; --ningun permiso a los roles públicos predeterminado
REVOKE ALL ON DATABASE postgres FROM PUBLIC; --ningun permiso a la base de datos
```

- Generar Rankeos: (Conocer casos de mayor acogida)

```
select nombre, num_pedidos, rank() over(ORDER BY num_pedidos DESC)
from cliente;
```

Data Output			
	nombre character varying (40)	num_pedidos integer	rank bigint
1	Mafe Naranjo	8	1
2	Martha Guacaneme	6	2
3	Sofia Arce	3	3
4	Luz Muñoz	3	3
5	Marisol Puentes	3	3
6	Laura Robles	2	6
7	Sara Villalobos	2	6
8	Angela Rojas	1	8
9	Ana Sofia Gutierrez	1	8
10	Camila Muñoz	1	8
11	Carolina Ardila	1	8

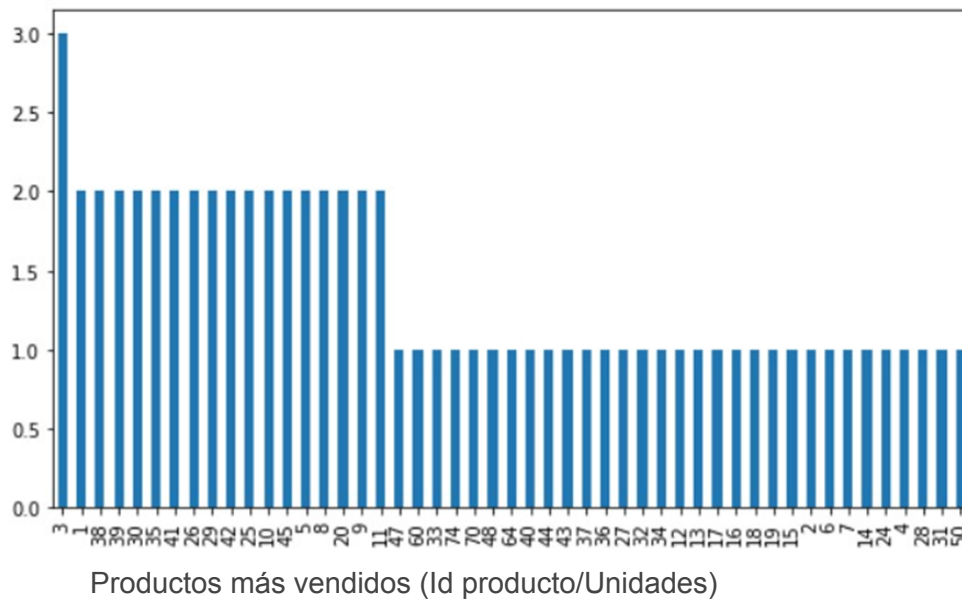
- Crear Triggers (Herramienta para realizar cambios automatizados dentro de la base de datos):

```
CREATE OR REPLACE FUNCTION restar_existencias()
RETURNS trigger
AS
$$
BEGIN
    IF EXISTS (SELECT * FROM producto p WHERE p.id_producto = NEW.id_producto) THEN
        UPDATE producto
        SET existencias = existencias - NEW.unidades
        WHERE id_producto = NEW.id_producto;
    ELSE
        RAISE EXCEPTION 'El id de producto no existe %', new.id_producto
        USING HINT = 'Check the id again';
    END IF;
    RETURN NEW;
END;
$$
LANGUAGE plpgsql;

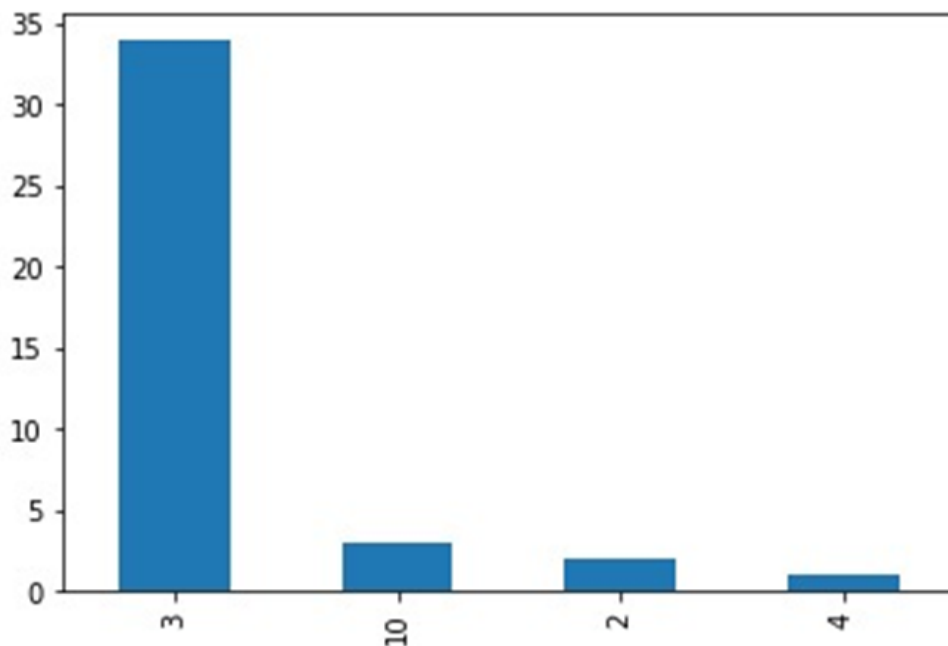
CREATE TRIGGER ventas AFTER INSERT ON factura
FOR EACH ROW
EXECUTE FUNCTION restar_existencias();
```

También una vista gráfica que nos permita conocer el comportamiento de la base de datos:

- Gráficas de barras para entender de mejor manera a los clientes o las ventas:



- Gráfica de barras para conocer la procedencia de los clientes por ciudad:



## Conclusiones:

1. Podemos comprobar que el modelo logra solucionar los inconvenientes que tenía el negocio, la dueña de (R&B) tiene un nuevo sistema de inventario eficiente, ahora maneja factura para saber todo sobre sus clientes, conoce el dinero que ingresa a su tienda y el que debe salir para volver a surtir su tienda.

2. Ahora la dueña de (R&B) sabe en qué ciudades tienen más clientes, desde que red social tiene mayor ganancia, cuales son sus productos más vendidos, sus clientes frecuentes y hasta forma de saber que productos debe surtir con urgencia.